REMARKS

Applicants respectfully request consideration of the subject application as amended herein. This Amendment is submitted in response to the Office Action mailed July 6, 2005. Claims 1-39 stand rejected. In this Amendment, Claims 1-8, 15-24, and 33-39 have been amended. Applicants submit that no new matter has been added by this Amendment.

Claim Objection

The Examiner objected to claims 36-39 for minor informalities. Claims 36-39 have been corrected.

35 U.S.C. §112, second paragraph

Claims 33-39 are rejected under 35 U.S.C. §112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the inventions for the term "logic block." Claims 33-39 have so been amended and Applicants submit that claims 33-39 no longer are indefinite for failing to particularly point out and distinctly claim the subject matter which Applicants regard as the invention.

35 U.S.C. §101

Claims 1-7, 15-23 and 33-39 are rejected under 35 U.S.C. §101 because the claimed invention is being directed to non-statutory subject matter. Claims 1-7, 15-23 and 33-39 have been amended to direct the claimed invention to a statutory subject matter

35 U.S.C. §103(a)

Claims 1-39 are rejected under 35 U.S.C. §103(a) as being unpatentable over Java 2, ("Java 2: The Complete Reference," hereinafter "Java"). Applicants respectfully submit that in the currently amended form, Java does not make obvious claims 1-39.

Independent claim 1 currently recites: (emphasis added)

A <u>computer implemented</u> method of customizing a product comprising:

providing a set of one or more customizable product classes;

receiving a request to designate a customizable product from the set of one or more customizable product classes as a customizable product instance;

presenting a customizable product class from the set of one or more customizable products classes as the customizable product instance;

providing a set of one or more component products, the set of one or more component products associated with the customizable product instance;

receiving a request to associate a first component product from the set of one or more component products with the customizable product instance; and

determining whether to associate the first component product from the set of one or more component products with the customizable product instance based on a predetermined set of customizable class rules.

Applicant submits that Java did not teach, suggest, or even motivate the limitation "determining whether to associate the first component product from the set of one or more component products with the customizable product instance based on a predetermined set of customizable class rules."

As taught by Applicants' Disclosure, as the number of components making the customizable product increase, the more complex the decision making process is for the consumer to determine which components to select. The computer implement method as claimed in claim 1 includes a determination of whether to associate the first component product from the set of one or more component products with the customizable product instance based on a predetermined set of customizable class rules. Additionally, the customizable rule may be designed by configuration experts, defining conditions and triggering actions between the component products that ensure the consumer purchase optimal solution that meets their needs. (Specification, page 6, [0025]). In one embodiment, such feature allows the configuration expert to tailor the user interface of component products to the customizable product. The problem sought to address by claim 1 was not taught, suggested, or motivated in Java.

Applicant respectfully disagrees with the Examiner's reasoning on even though Java does not explicitly disclose the step of determining whether or not to associate the first or a second

component product with the customizable product instance that Java disclose inheritance as a cornerstone of object-oriented programming because it allows the creation of hierarchal classifications of subclasses and therefore it would have been obvious to one of ordinary skill in the art to allow a user to determine whether or not to associate a first, second, or third component product with the customizable product instance. The teaching of Java may have taught customizing a class or a product class, even if based on inheritance. Java only taught that to inherit a class, you simply incorporate the definition of one class into another. This teaching is not the same or even suggesting as determining whether to associate the first component product from the set of one or more component products with the customizable product instance based on a predetermined set of customizable class rules.

Claims 2-7 depend indirectly or directly from claim 1. Independent claims 8, 15, 24, and 33 and their respective dependent claims contain at least similar limitation as claim 1 previously discussed. Therefore, for at least the reasons above, Applicants respectfully submit that Java does not make obvious claims 1-39.

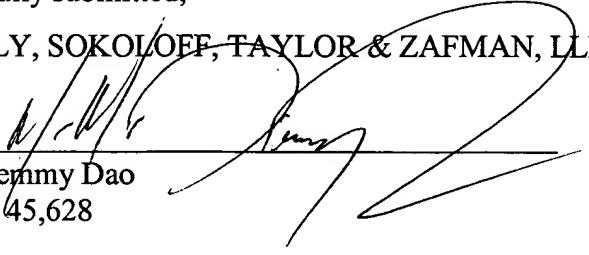## DEPOSIT ACCOUNT AUTHORIZATION

Authorization is hereby given to charge our Deposit Account No. 02-2666 for any charges that may be due. Pursuant to 37 C.F.R. 1.136(a)(3), applicant(s) hereby request and authorize the U.S. Patent and Trademark Office to (1) treat any concurrent or future reply that requires a petition for extension of time as incorporating a petition for extension of time for the appropriate length of time and (2) charge all required fees, including extension of time fees and fees under 37 C.F.R. 1.16 and 1.17, to Deposit Account No. 02-2666.

If the Examiner determines the prompt allowance of these claims could be facilitated by a telephone conference, the Examiner is invited to contact Marina Portnova or Mimi Diemmy Dao at (408) 720-8300.

Respectfully submitted,
BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN, LLP

Dated: October 6, 2005

Mimi Diemmy Dao
Reg. No. 45,628

12400 Wilshire Boulevard
Seventh Floor
Los Angeles, CA 90025-1026
(408) 720-8300